Preliminary Copy

A Graphical User Interface to Prepare the Standard Initialization for WRF

NOAA Technical Memorandum
To Be Published September 2003

Paula McCaslin, John Smart, Brent Shaw, Paul Schultz, Steve Albers, Dan Birkenheuer, John McGinley and Linda Wharton
NOAA/Forecast System Laboratory
Boulder, Colorado

1.0 Introduction

A graphical user interface (GUI) is a commonly used computer tool that allows a user to more easily interact with underlying software and files. Though the GUI itself can be quite involved and contain several levels of interactivity, it ultimately allows a safer and more efficient method to manipulate the software and modify files. A great advantage is that users do not have to remember the file paths or executable names, and therefore they do not have to remember their interdependencies. The GUI also prevents users from making potentially serious mistakes or becoming confused about the intricacies of the system software.

The Weather Research and Forecasting (WRF) system is being developed with the latest scientific knowledge and computer technologies. Several organizations are participating in this development and are using established software as well as writing new software as the system evolves. The Forecast Systems Laboratory (FSL) has developed the SI system (Ref) an important and necessary first step in setting up WRF. The SI provides two mandatory parts of WRF:

1. To define and localize a three dimensional grid; hereafter this is referred to as domain localization; and,
2. To provide the initial and lateral boundary condition files by interpolating larger scale model data to the domain.

The SI is designed to allow significant flexibility achieved using three primary directory structures: one for the source software, one for the installed executables and one for the output data. These are further described in section 3.0. The purpose of this document is not to describe the details of the SI since this can be obtained from other documentation (see references); on the other hand, a general understanding of the system is required and knowledge of the three-directory structure is important.

This initial step in preparing SI presumes that a user has obtained and installed the software. This means the executable programs have been successfully downloaded (from the SI tape archive (tar) file via the WRF web site: http://wrf-model.org/si) and compiled. Completing an implementation requires that users have some understanding of the SI system design and interdependencies. Installation is manual and instructions are detailed in the README information file found in the SI tar file.

The primary objective of this report is to describe the WRF GUI and provide an understanding of its functionality. The GUI runs all SI processes to generate domain localization, initial and lateral boundary condition data and display graphics.

Details of the GUI software and how to install it are presented in section 2.0. Section 3.0 describes its design and layout and gives an example of localizing a domain over Australia. Future work is discussed in section 4.0. A brief summary is found in section 5.0.

1.1 Domain Localization

The SI satisfies global domain localizations in the commonly used map projections: Lambert Conformal (both tangent and secant), Mercator, and Polar Stereographic. Some basic understanding of map projections is valuable but not mandatory. The phrase domain localization means to complete the following tasks:

- Prepare a directory structure where the WRF data will be written.
- Choose a rectangular domain on earth, with a fixed center point and projection type.
- Edit the domain for the specific number of grid points, grid spacing, standard latitude and longitude values to insure the resulting map covers the desired area with this set of parameters.
- Determine the distribution of vertical levels.
- Set the paths to geography data including: terrain, land-use, greenness fraction, albedo, soil type, and global mean temperature.
- Run the localization Perl scripts which, in turn, run the localization Fortran software.
- Verify that the localization is correct with generation of graphics images.

1.2 Interpolation of Data

In addition to domain localization, SI also includes software that prepares large-scale model output for WRF model initialization and lateral boundary conditions. This step requires successful completion of domain localization, which, in turn is a condition that influenced the GUI design.

2.0 Software

2.1 Selecting Perl/Tk

Perl is the scripting language used to configure WRF. It is a powerful software tool automating many aspects of the software setup. These scripts, however, have environment and file variables that need to be set prior to running them as well as setting many command-line arguments. This can be a complex task for users. The idea for a graphical user interface was an obvious solution to accomplish this task efficiently and accurately. Many GUI languages and tools were considered and evaluated. The GUI was written Perl/Tk. One key motivation for this choice was compatibility with the existing SI scripts. Key justifications for this choice follow.

Perl is arguably the most popular scripting language in use today. It is used for a wide variety of tasks, including file processing, system administration, web programming and database connectivity. Early Perl users had to be content with command-line interfaces. But the splitting-off of the Tk widget library from the Tcl language opened a whole new world to Perl. Perl programmers could now easily create graphical interfaces for their programs using Tk's flexible and friendly widget set and, with little effort, those programs could be made to work across Windows and Unix platforms.

The relatively recent advent of the web browser would seem to have made the Tk interface obsolete. CGI programmers are almost inherently cross-platform and provide many of the same widgets as Tk (this includes Menus, Buttons, text entry fields, and so on). However, the inherent statelessness of the Web makes it difficult to write some programs for it. Perl/Tk provides a richer widget set than available to the CGI programmer. Server push and client pull try to get around some limitations, while JavaScript fills in other gaps, but the fact is, the user experience still fall short in many instances. It is for precisely this reason that Perl/Tk continues to flourish.

The Tk module gives the Perl programmer full access to the powerful Tk widget set. This rich and diverse library, like Perl itself, makes the easy things easy and the hard things possible. [LIDI02]

Perl/Tk programs are written in an object-oriented (OO) style. Perl/Tk widgets (such as Buttons) are objects that have methods we use to control them. Besides widgets, Perl/Tk has images, which are also objects. Each different widget belongs to a class. A widget's class defines its initial appearance and behavior, but individual widgets of the same class can be customized. As an example, we could create two Buttons that have different textual labels but are otherwise identical. The widget class constructor instances a widget. The class also defines a widget's initial behavior by creating bindings. A binding associates an event such as a button press with a callback, which is a subroutine that handles the event. [LIDI02]

Using Perl/Tk allows the user to call Perl module subroutines directly. To contrast, choosing another user interface language would add a layer of communication between two different languages -- a step not necessary using Perl/Tk.

Further motivation in selecting Perl/Tk is that it is free, platform independent without modification or recompiling and it is open source software that is very well supported in the community.

2.2 Other software used in GUI (Perl, C, Fortran)

As mentioned, Perl is the scripting language used to configure the SI. Many Perl scripts directly edit, modify, and create files and directories used by the WRF system and its processes. There are several Perl scripts invoked from the GUI by simply pushing a button. For example, window_domain_rt.pl is called from the GUI with all the necessary command-line arguments resolved by the GUI.

The projection mapping software is written in C. The C compiled software allows for faster code than would be possible in Perl since it is, generally, interpreted code. One approach than might be considered in the future is to extend Perl by creating a C library that Perl can understand and link with. Forgoing this, we currently execute C code from Perl, which generates and writes output files to /tmp that are subsequently read as input to Perl/Tk.

The SI software is written in Fortran, Fortran90 and C. It is fully dynamic by virtue of the Fortran namelist entries for domain size and configuration. Much of the software has roots from the FSL Local Analysis and Prediction System, (LAPS). LAPS has been in existence for many years and the grid localization software is particularly robust. It is used by many researchers, operational weather centers and organizations world-wide and has gone through numerous

2.3 Installing

The Tk extension does not come with the standard Perl distribution. You will need Perl/Tk version 800.023 or higher. If this is not on your system it is easy to install by visiting the CPAN (Comprehensive Perl Archive Network) at www.cpan.org/authors/id/NI-S and look for the version in the form of Tk800.023.tar.gz. It is recommended that your system administrator install Perl/Tk on your system.

During installation of the SI the user is asked, “Would you like to install the WRFSI GUI?” If the response is ‘Yes’ and Perl/Tk cannot be found, then the Perl/Tk will be installed from the tar file ported with SI version 1.3.2.

The location for WRF GUI source code is within the SI source code directory, specifically in SOURCE_ROOT/gui.

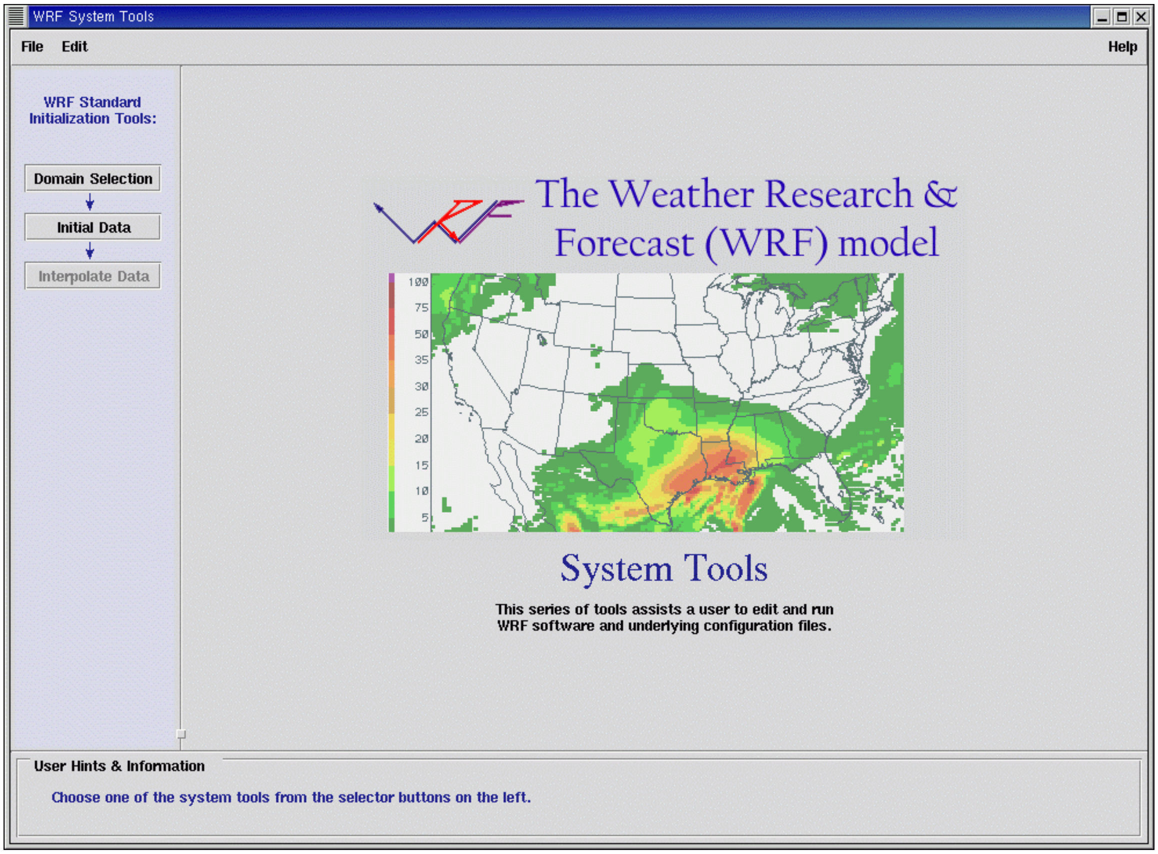

Figure 1: Display of WRF GUI application at start-up.

3.0 Layout and Design

The GUI described here is customized for WRF and a banner image is displayed upon start-up (Figure 1). The GUI automatically determines a WRF realization through a simple evaluation of the system design and file structure. The GUI is capable of running other systems besides WRF and, as such, a different banner would be displayed. We allow this capability because other weather analysis and forecasting systems can use this GUI to configure and localize domains (in particular, LAPS).

The SI UNIX environment, in which the GUI runs, allows the source software directory to be located in one area (hereafter, SOURCE_ROOT), the executable software to be located in a second directory (hereafter, INSTALLROOT) and the I/O directory in a third area (hereafter, DATAROOT). The nature of this design allows full flexibility of the system but it also adds a degree of complexity making it difficult to set up and initially run. Experienced users are more likely to succeed on the first try; however, it is typical for users that fail on the first try to give up. The three-directory structure approach is implemented in SI by using environment variables. Because the GUI is delivered with the tar file, the SOURCE_ROOT is known (i.e. the location in which the tar file was "un-tarred"). Users run the GUI from the INSTALLROOT so this is also known. In a typical SI implementation the SOURCE_ROOT and INSTALLROOT share the same directory path name.

3.1 Application Layout

The design of the WRF GUI is to present graphical tools to completely run all of the SI processes in a robust manner that makes sense to the users of the application. Instead of making users read significant documentation to begin these tasks, the GUI provides partially and fully completed solutions (e.g. filled in entry box values) for users to edit, if need be, to satisfy their objectives. Users can bypass very complicated procedures, especially for those unfamiliar with scripting. They can get useful results without otherwise knowing how to proceed from scratch.

The application layout consists of process buttons from the tool selector to the left of the window and a display containing the editing tool to the right of the window. Always present in the GUI window are the standard menubar at the top of the window. The user can exit the application or query the help pages and version information by selecting options found on the File and Help buttons of the menu bar, respectively (upper left and right corners). We also find at the bottom of the GUI window a ‘Hints & Information’ area sometimes suggesting steps to be taken and other times summarizing the success of the step that has been completed. See Figure 1.

There are several components to SI and a specific sequence of processes that take place. The buttons of the tool selector (on the left side of the window) with its arrows direct the user in moving through these steps. The steps are: Domain Selection, Initial Data, and Interpolate Data. Once a user presses a button from the tool selector, a corresponding editing tool is displayed. These are described in detail in sections 3.2 – 3.4.

Displaying a lot of information in a small space is a big challenge in a GUI application. The tabbed panel approach, where tabs are located across the top of each panel, is designed to address this. A tabbed panel for Domain Selection’s editing tool interface is shown in Figure 2. It allows the application to have many panels of information, but allows only one to be shown at a time. An additional advantage of the tabbed panel approach is sequential navigation.

The over-all design includes suppressing detail and complexity until the user needs it. As such, some controls are disabled and others are not displayed until they are relevant to the user accomplishing a task. Tool tips (small information boxes) are also displayed when the user places the cursor on certain widgets within the application.

The GUI design was influenced by applications that FSL evaluated during this development, including: the Air Force Weather Agency (AFWA) System GUI and the National Center for Atmospheric Research (NCAR/ARMY) MM5 GUI. (We’ll need some references here).

3.2 Domain Selection Tool

The Domain Selection Tool has the following panels: Domain, Horizontal Grid, Vertical Grid, Localization Parameters, Localize Domain and Domain Graphics. The order in which the panels appear reflect the order in which the steps of localization need to be performed, starting from the left and proceeding to right. Below the panels are two control buttons: '<Back', and 'Next>' for the user to navigate step-by-step through an otherwise complicated process with instructions and choices specified as needed.

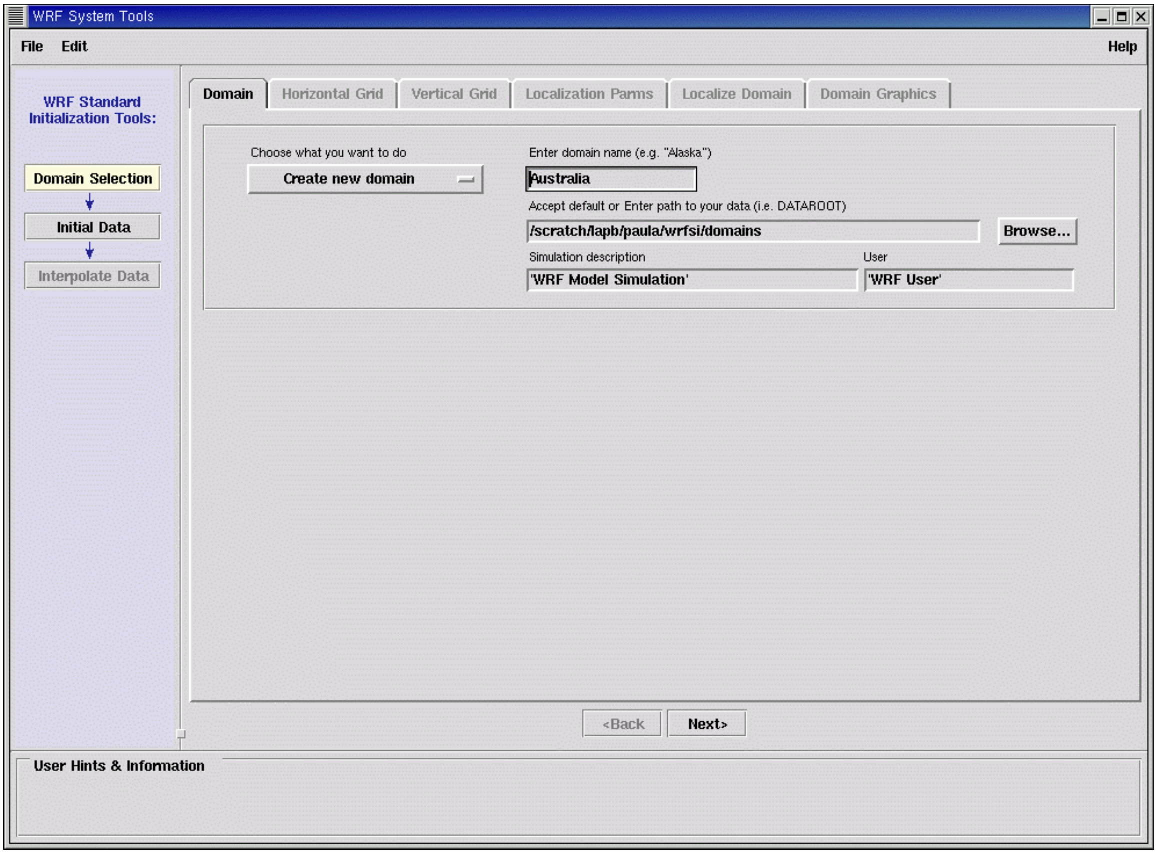

Figure 2: Display of Domain Selection Tool, showing the Domain panel.

3.2.1 Domain Panel

The Domain panel initiates the first task of the localization process; this is to declare a domain.

There are four choices available on the pull-down menu shown near the top of the panel allowing the user to create, load, copy or delete a domain. Once a domain is initially created, it can subsequently be edited, copied or deleted (Figure 3). The GUI application includes a default domain, called 'default', to serve as an example of steps in the process of domain localization. This particular domain cannot be edited or deleted only copied then edited.

The top of the Domain panel (Figures 2-4) contains four entry fields: domain name, path to your data or DATAROOT (with a file Browser widget), simulation description and user description. The user can accept default values suggested by the GUI or enter new values by typing them in. Users typically enter useful names for their domain; in this example an area encompassing Australia will be localized and 'Australia' was entered as the domain name. A disabled entry signifies that an entry or button is un-editable as a function of the selected mode (e.g. copy existing domains).

Once a domain has been created and localized the GUI generates files that reference its DATAROOT. Because the DATAROOT directory contains the localization results and model output, the directory path entered must have ample disk space (or a disk partition). The environment variable called "MOAD_DATAROOT' is actually the DATAROOT/domain name. DATAROOT and MOAD_DATAROOT are synonymous in this paper.

We limit the discussion to new localizations; however, the GUI allows modification or review of existing domains. This is accomplished by selecting the option 'Choose what you want to do' from the pull-down menu shown on the Domain panel.

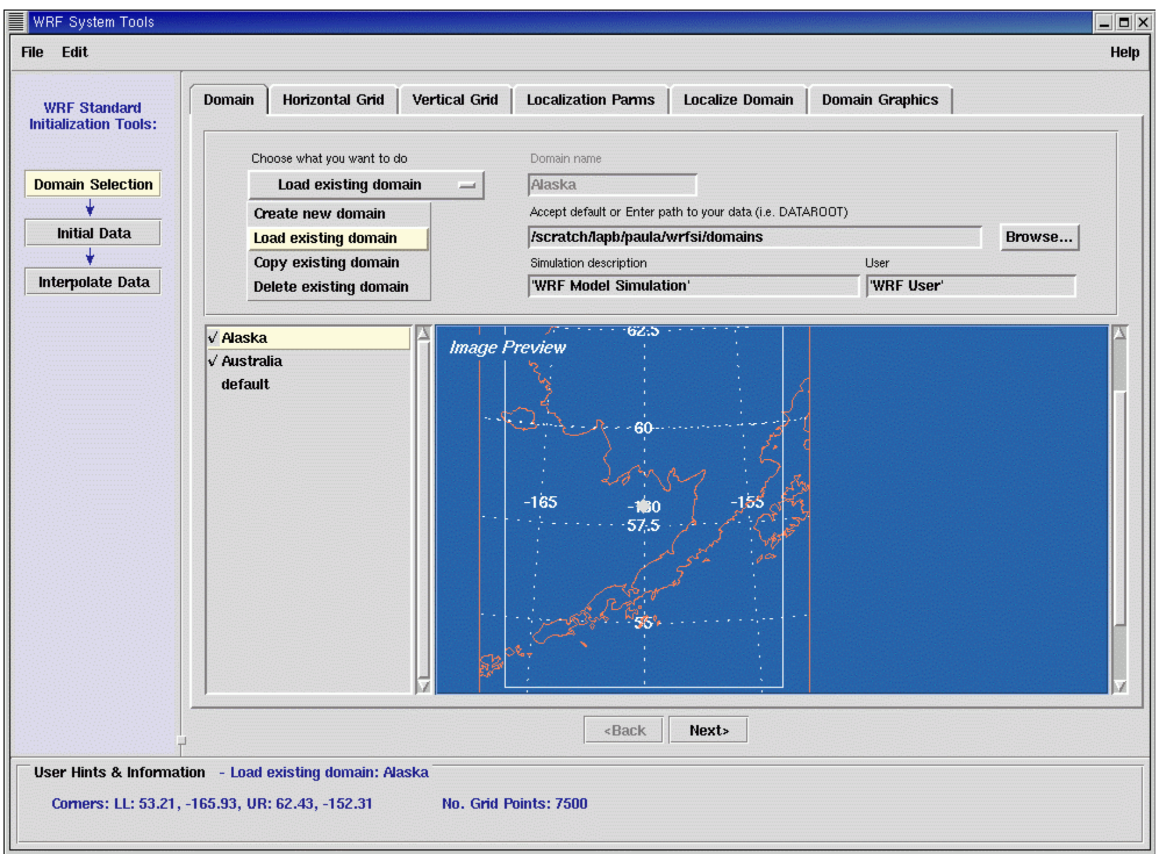

Figure 3: Display of Domain panel showing "Load existing domain" mode. Notice that there is a preview image of the domain.

The domain selection choice ‘Load’ will load an existing domain into the GUI. Everything about the domain can be edited except for the disabled domain name.

The domain selection choice ‘Copy ’ will create a new domain from an existing one. The automatic name for the new domain is generated from the original name adding a case number. For example, if Alaska were selected by the user to be copied its automatic domain name would be Alaska_001 for the first domain copy, with a limit of 999. The user is free to edit any parameter for this copied domain including its name.

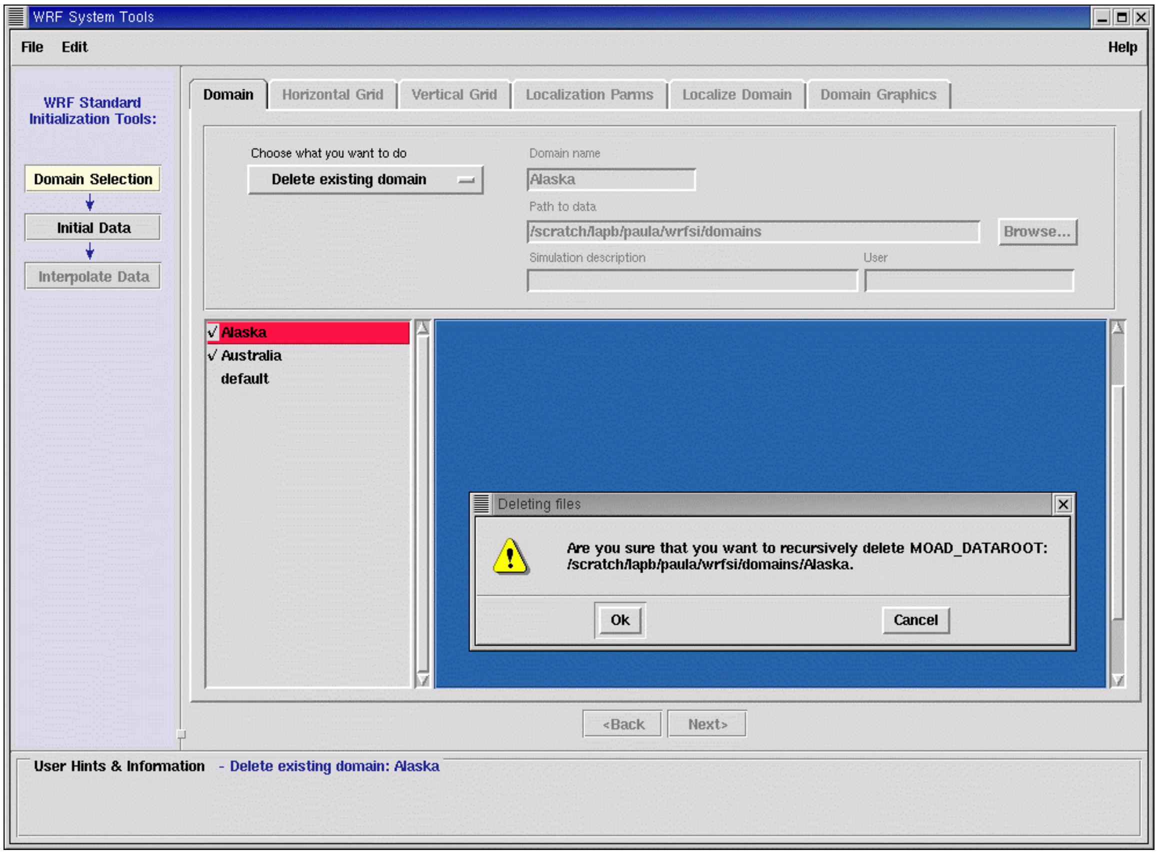

Figure 4: Display of Domain panel showing "Delete existing domain" mode.

The domain selection choice ‘Delete’ will highlight the domain selection in red then asks the user to confirm the action before a recursive remove is performed.

3.2.2 Horizontal Grid Panel

The Horizontal Grid panel (Figure 5) is comprised of two panels containing tools that allow the user to define a new model domain by drawing a bounding box on a map (left panel) and editing map projection variables (right panel). The map window initially shows a Cylindrical Equidistant projection map of the world centered at Latitude 0 and Longitude 0. The entire global map is not displayed, but the panel has a sliding scroll bar on the bottom and right sides to reposition it in the panel. The global map can be increased or decreased in size by pressing the appropriate zoom-in or zoom-out icon buttons just above the map.

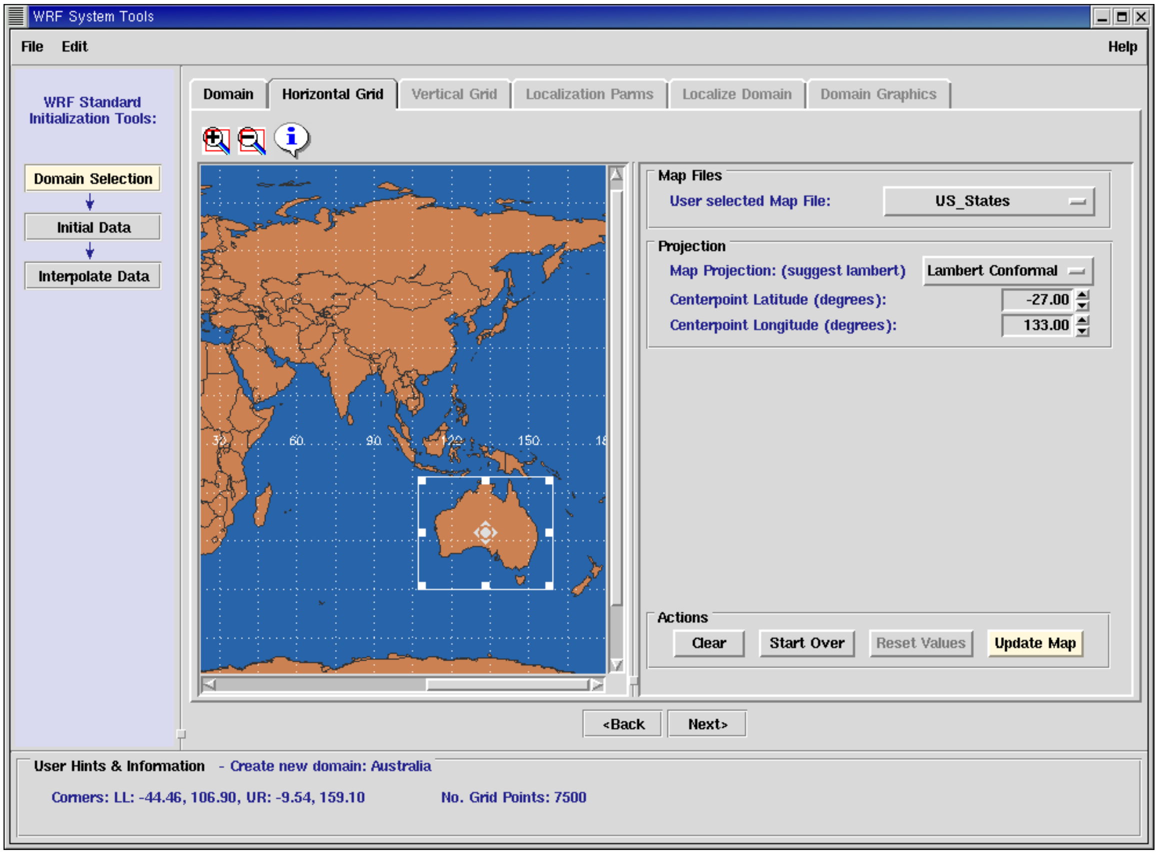

Figure 5: Display of Horizontal Grid panel.

As the user presses mouse button 1, on the global map, and drags the mouse to a new location on the map a white domain bounding box will be drawn to specify the domain. This was done for the area over Australia seen in Figure 5. The process of creating a domain bounding box, then moving and resizing the bounding box is described for the user by pressing the information icon button '**i**' located just above the map. Experimenting with pressing mouse button 1, dragging the cursor and pressing the 'Clear' button is also a good way to learn the functionality of creating a domain bounding box and fine-tuning its location and size.

Once a domain bounding box is created status information is displayed in the Hints & Information panel, specifically, the lower left (LL) and upper right (UR) corner latitudes and longitudes of the domain bounding box and the total number of X, Y grid points in the selected domain.

The top sections of the right control panel are used to set a background map, the projection type, and the center point values. As the domain bounding box is moved or resized, the values in the right control panel will update. Likewise, the user can manually edit the center point values in the right panel and the domain bounding box will update.

The GUI has built in criteria to assist the user in chosing a projection based on domain's the center point latitude. The 'Map Projection' label suggests Lambert Conformal based on center latitude of 63 degrees, but Polar Stereographic could be chosen instead (see Figure 5).

At the bottom of the right control panel are four action buttons: Clear, Start Over, Reset Values and Update Map. The Update Map button will draw a new map with the listed projection parameters. The Start Over button will undo all updates to the domain map but will leave the original domain bounding box. The Clear button resets all widgets and deletes the domain bounding box. The user can fine-tune the domain bounding box and press 'Reset Values' if these changes are not acceptable and to press 'Update Map' if they are.

A warning dialog box that will appear suggesting steps for the user if parameter values are not chosen prior to pressing the 'Update Map' or 'Next>' buttons. Notice that 'Update Map' is highlighted yellow (Figure 5) suggesting this that button needs to be pressed. Any change to a parameter will cause the 'Update Map' to highlight indicating that the map's image and the projection parameters are out of sync with each other. One of two things needs to happen, either press 'Update Map' to draw a new map, or press 'Reset Values' to return to the current maps parameters to the entry boxes.

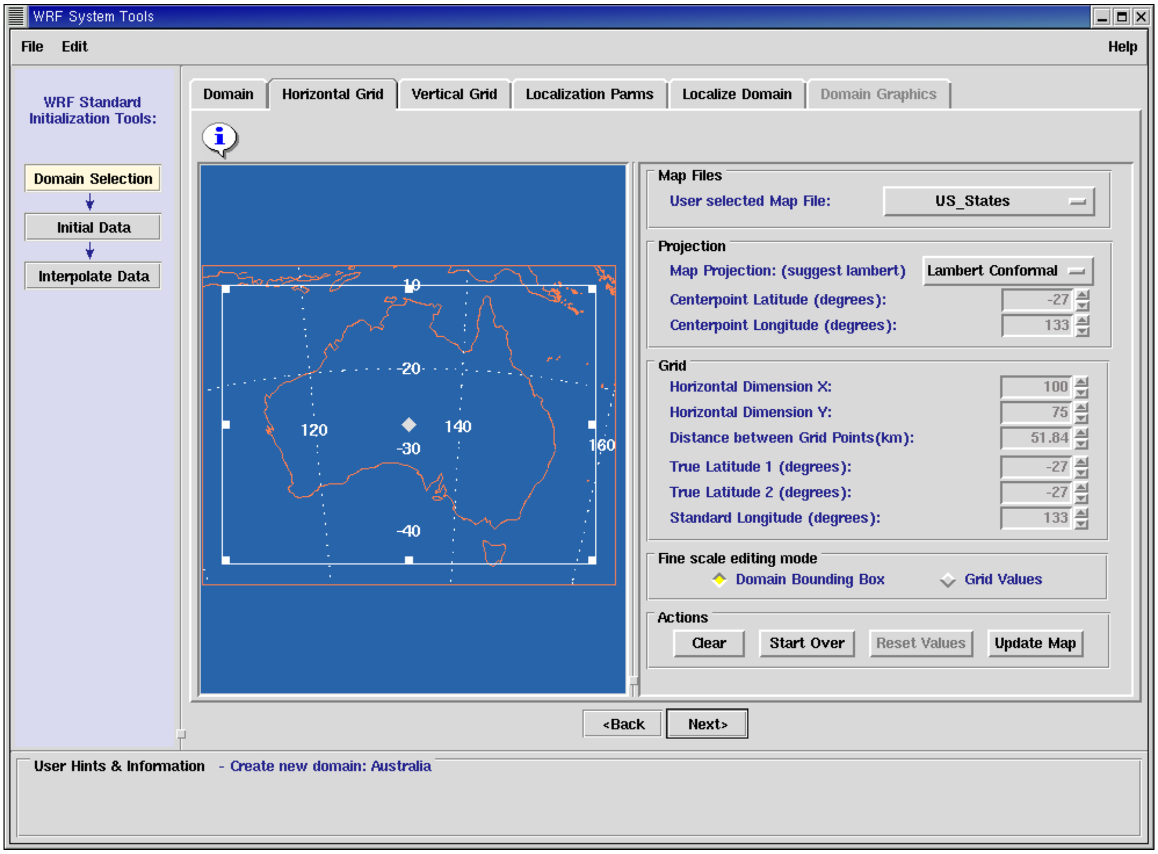

Figure 6: Display of Horizontal Grid panel following the user action of pressing 'Update Map'.

When the user presses 'Update Map' both left and right panels of the Horizontal Grid editor update as seen in Figure 6. The domain is now displayed in projection space (in this example, Lambert Conformal) based on user-selected information. You may notice that the lower left and upper right corner points have changed slightly due to the transformation from the previous map to the new gridded map in projection space. The new corner points are accurate. To obtain this gridded map, we first assume that 'Horizontal Dimension X' equals 100. The great circle distance between the east and west sides of the domain is computed which, in turn, is used to estimate 'Grid Spacing'. Similarly, computing the north-south great circle distance and multiplying this by the grid spacing from the previous step determine 'Horizontal Dimension Y'. Additional editing options are available on the right control panel. These allow for exact adjustments to the map. (The staggered grid when applied in gridgen_model will affect the lower left and upper right corner points by slightly changing these values further.)

It is common for the user to refine the domain bounding box. There are two choices to do fine scale editing: the domain bounding box or the grid values. The 'Domain Bounding Box' mode allows the user to interactively resize the domain bounding box using the cursor to manipulate the white bounding box in the left panel (Figure 6) while keeping the center point and grid spacing values fixed. The 'Grid Values' mode allows the user to enter values to adjustment the domain bounding box (Figure 7). When the Grid Values is selected the domain bounding box looses its tags to inforce that the box is not interactive.

When the user is satisfied with the domain's map they proceed to the 'Vertical Grid' panel by pressing on 'Next>'. This step automatically creates a template directory and writes or updates the wrfsi.nl, dataroot.txt and domain.gif files.

Figure 7: Display of Horizontal Grid panel with 'Grid Values' selected as the method to fine-tune the domain.

3.2.3 Vertical Grid Panel

The Vertical Grid panel (Figure 8) is comprised of two panels containing tools that allow a user to view sigma levels (left side) and edit these levels (right side). Included in the right control panel are several options to edit the sigma levels that update the display. One option allows the user to enter a desired number of levels and choose one of three sigma level scheme that automatically generate levels. The schemes are calculate linear levels in sigma, calculate square root levels in sigma, or calculate the top 1/3 of the requested levels in linear and the lower 2/3 of the requested levels in square root in sigma. A second option is to load an existing file. A third option allows the user to add or remove sigma levels via a text editing window. Status information is sent to the Hints & Information panel. It contains the minimum and maximum sigma height distance values, which are affected by the surface pressures and temperatures and the highest pressure level at the top of the domain. Vertical sigma levels can be displayed in log pressure with the press of a button.

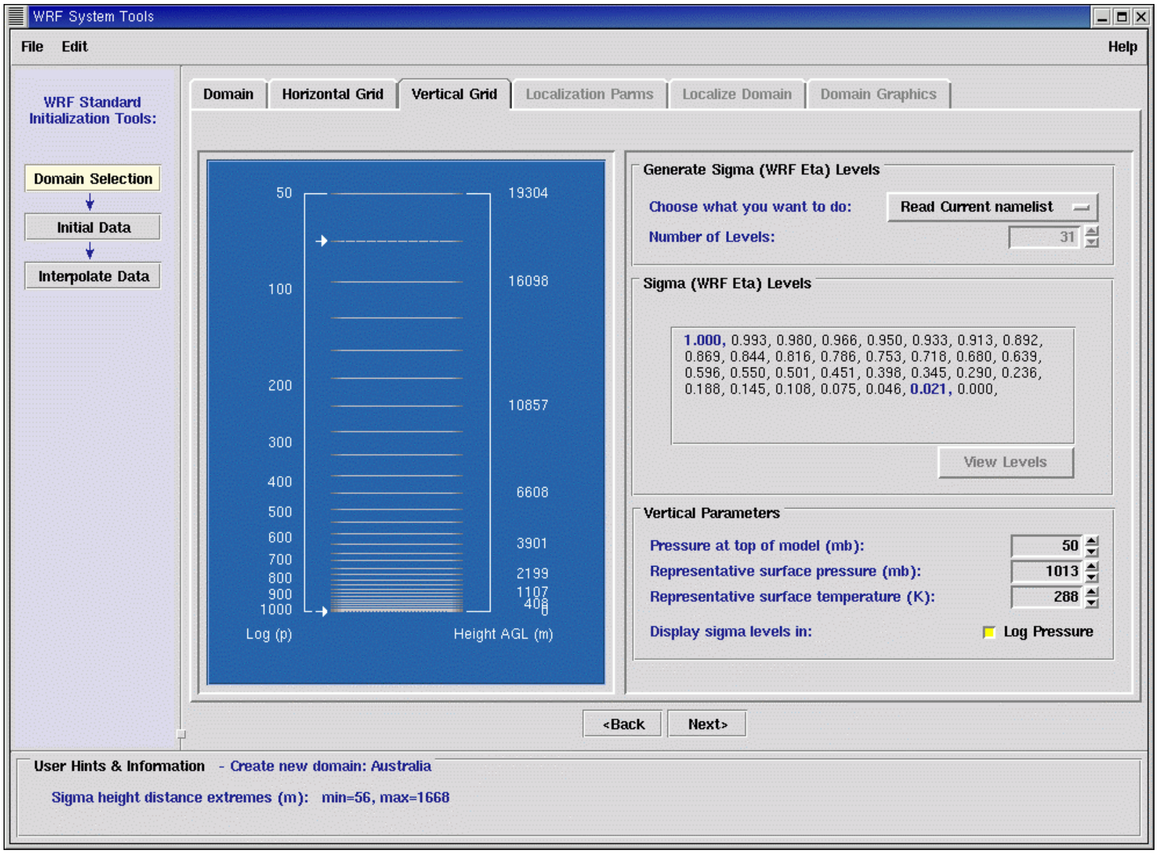

Figure 8: Display of Vertical Grid panel.

3.2.4 Localization Parameters Panel

The SI FORTRAN namelist file (called wrfsi.nl located in SOURCE_ROOT) is read in when the WRF GUI is started. From the beginning of the session to create or edit a domain the user selections and entered values change the variables in the domain's namelist. If the user is not satisfied with the values in the namelist, they should be edited. The paths to surfaces geography files are editable. Please refer to the SI README file (Appendix A) to learn more about the geography data and how to acquire it.

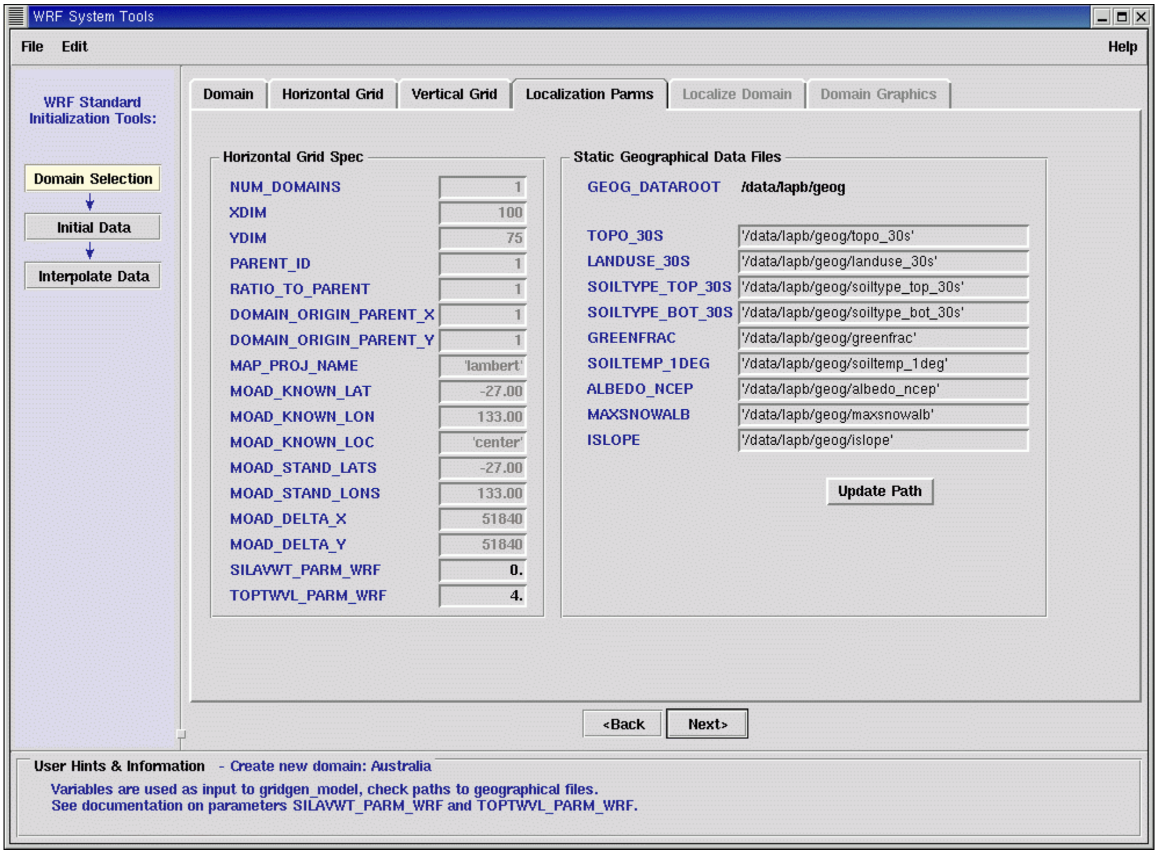

Figure 9: Display of Localization Parameters panel.

When the user is satisfied with the values in the previous and current Domain Selection panels the next step is domain localization.

3.2.5 Localize Domain Panel

At this point the user is ready to run the Perl and Fortran programs to localize the domain. The window_domain_rt.pl command and all its command line arguments are shown in a (non-editable) text window (Figure 10). To run this process, the user presses ‘Localize’. Depending on the grid size and grid spacing of the configured domain, this process can take seconds to several minutes to complete. Output from the process, otherwise sent to the screen, is redirected to the same text window mentioned.

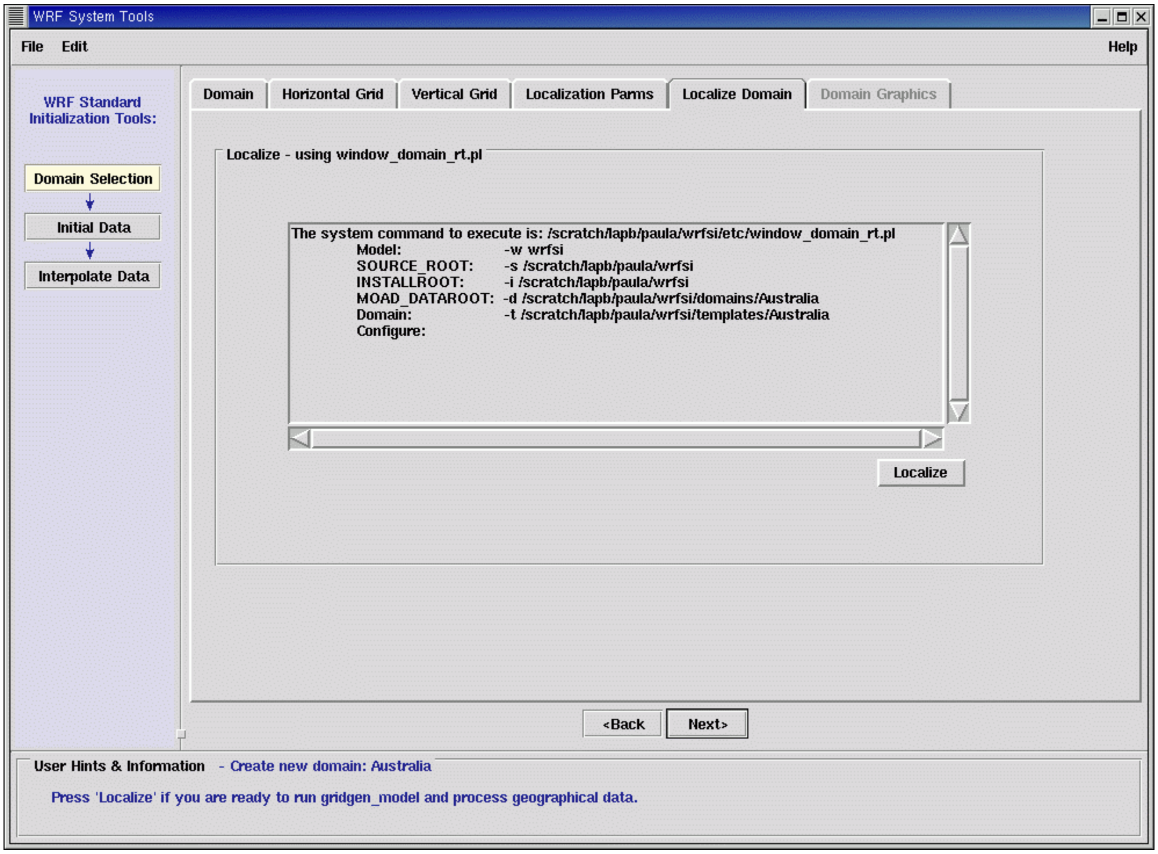

Figure 10: Display of Localize Domain panel.

The end product of domain localization is a defined and localized three-dimensional grid. Of specific interest is a netCDF (network Common Data Format) file in the DATAROOT called 'static.wrfsi' (hereafter, static file). The static file contains all non-varying information required for the WRF model, including terrain, land-use, latitudes and longitudes, map factor and several other quantities.

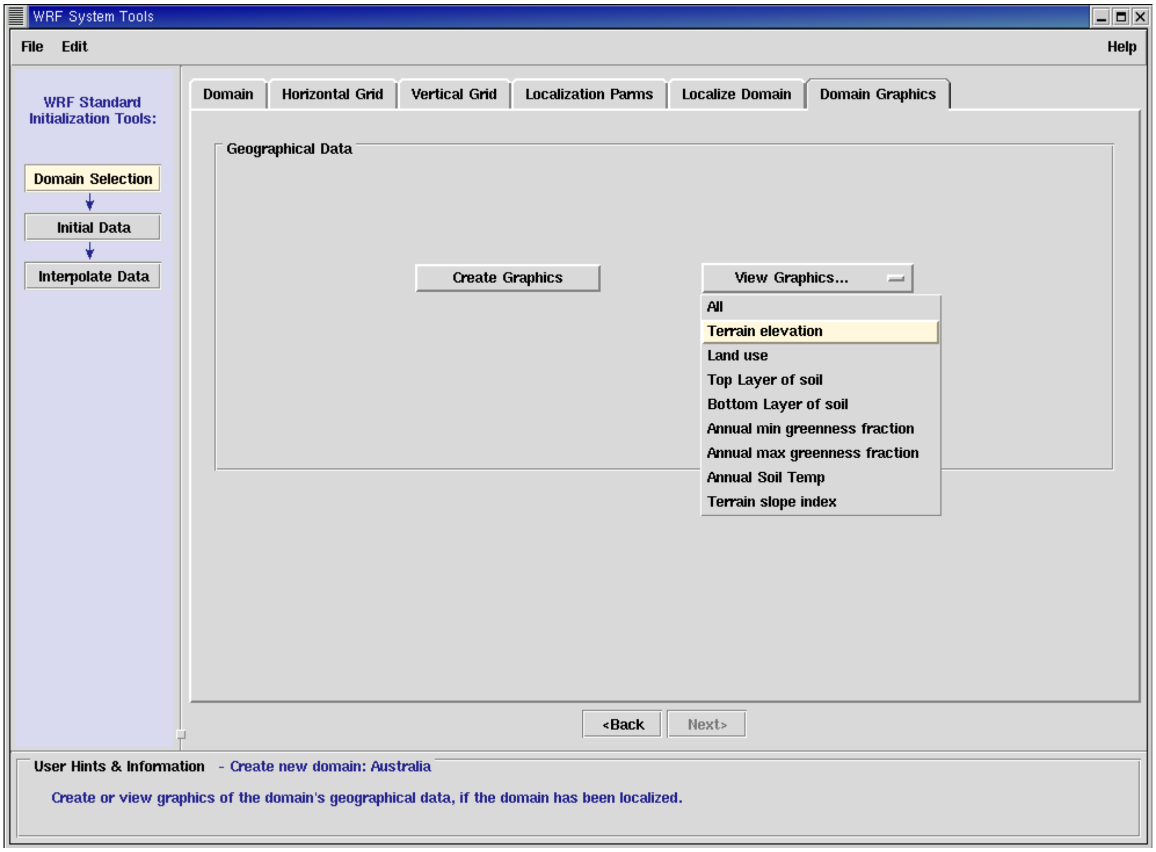

Figure 10: Display of Domain Graphics panel. This user has NCAR Graphics NCL available and as such, they are able to create and view output from domain localization.

3.2.6 Domain Graphics

There is a capability built into the GUI to create and display graphics of the localization results. Currently this is only possible if NCAR Graphics 4.3.0 is installed with NCARG_ROOT and NCL_COMMAND properly set. Example images of graphics output from the Australia localization include terrain (Figure 11), land use (Figure 12) and maximum greenness (Figure 13). These were generated by NCAR Graphics NCL and are viewable by NCAR Graphics idt launched from the GUI. The full graphic output (ncarg meta) files include the following variables:

1. Terrain.
2. Land-use.
3. Soil type.
4. Greenness fraction.
5. Mean annual soil temperature.
6. Terrain slope.

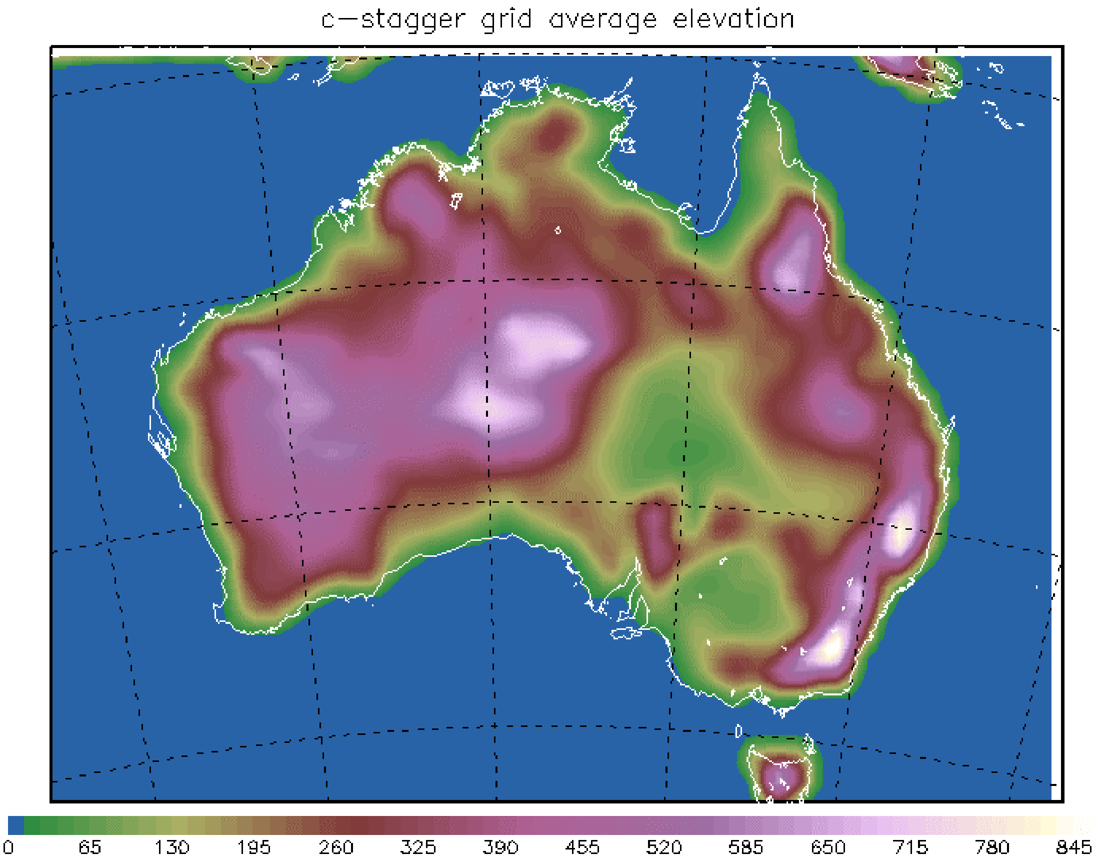

Figure 11: Image of terrain for the Australia domain localized in the example.

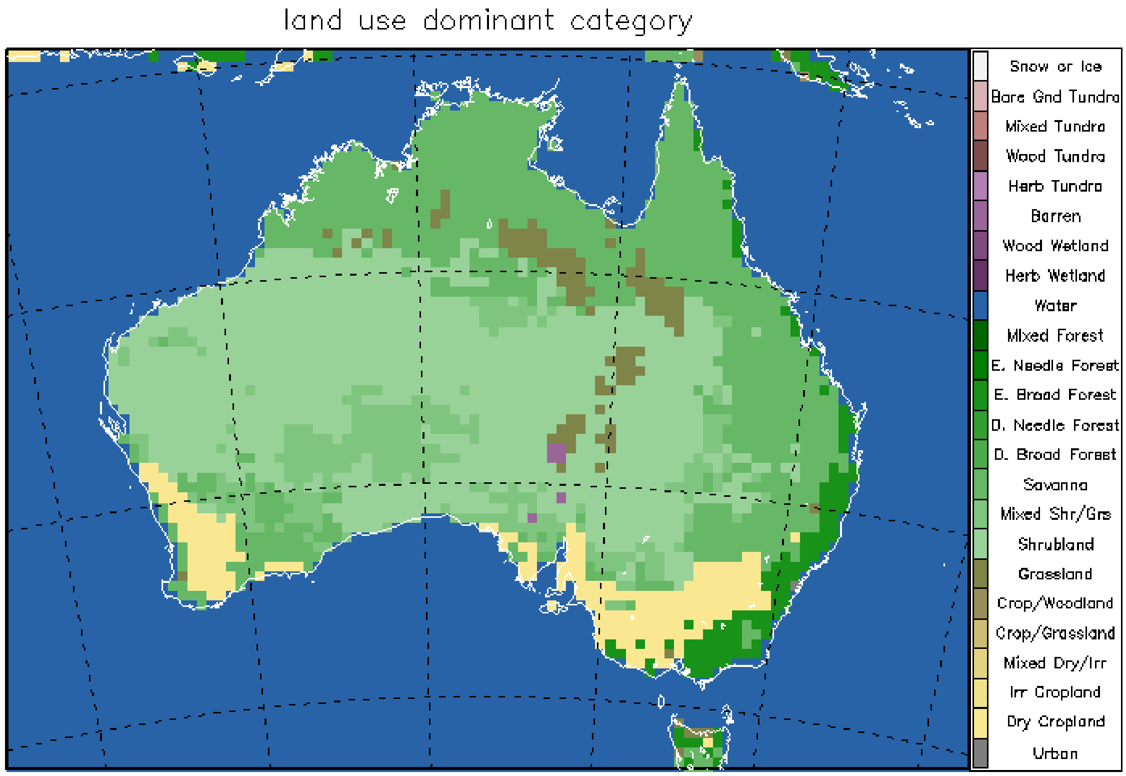

Figure 12: Image of land use for the Australia domain localized in the example.

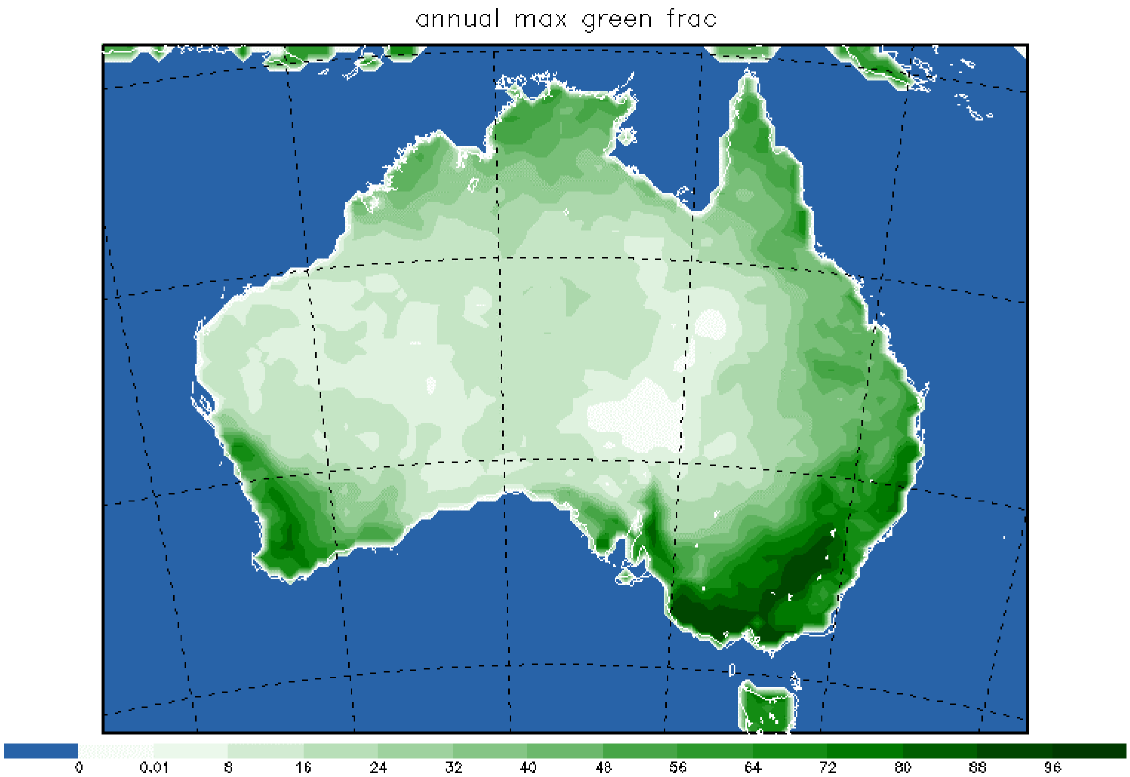

Figure 13: Image of max greenness for the Australia domain localized in the example.

3.3 Initial Data Tool

The Initial Data Tool performs the first of a two part task to provide the initial and lateral boundary condition data files. This section of the user interface runs a script (grib_prep.pl), which acquires background data then uses specified directories to decode and store this data.

Please note that the values entered in the Interpolate Data Tool in the 'Controls' panel (Section 3.4.1, Figure 17) need to correspond to choices made here, specifically, the values for ANALPATH and LBCPATH need to be considered.

3.3.1 Sources Panel

The *grib_prep Fortran executable* is responsible for decoding GRIB files. The grib_prep.nl namelist is a file that controls the execution of the grib_prep executable. This section of the GUI assists a user to edit their *grib_prep.nl file* for their system configuration.

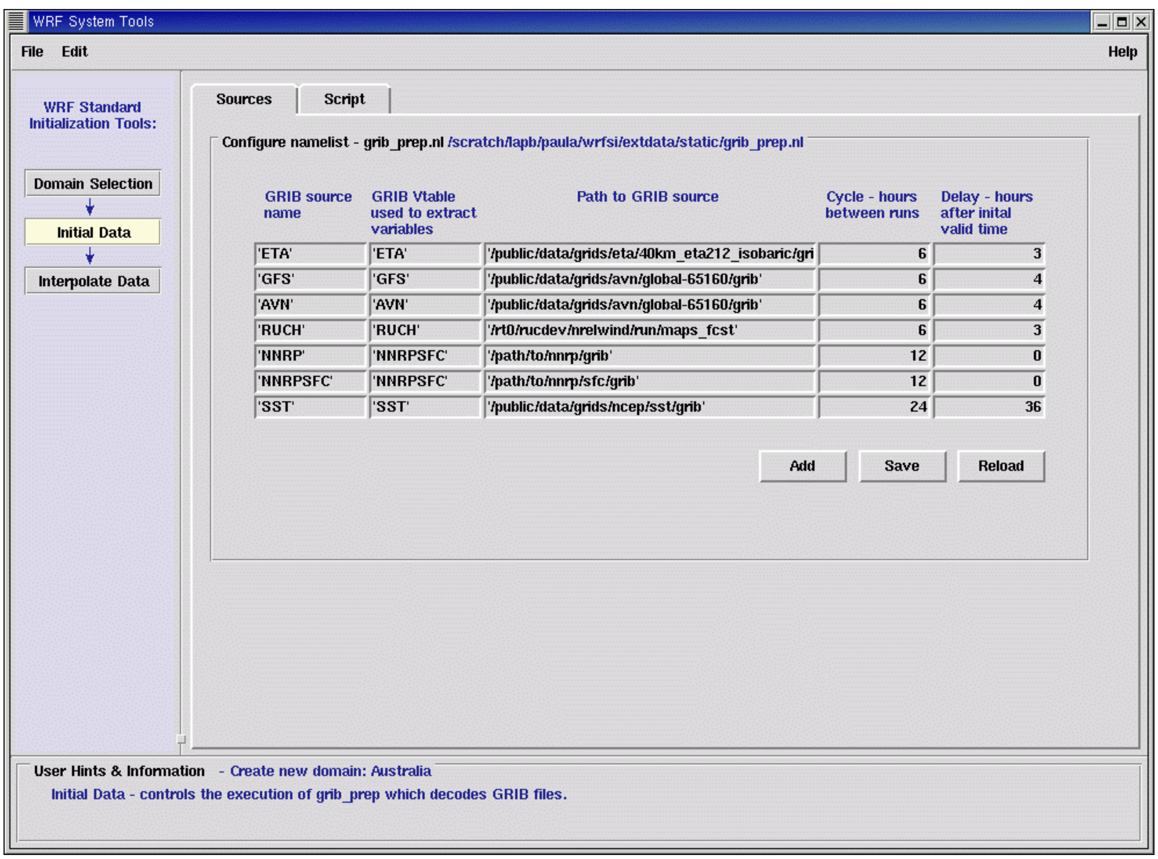

Figure 15: Display of Initial Data Tool - Sources panel.

As can be seen in the user interface (Figure 15) each entry line in the table of GRIB information is separated into five columns that need to be edited for the users system configuration. These values can be edited or deleted by using the keyboard. An entire entry line can be left blank but for the underlying software to be successful, no values on a valid line should be left empty. To delete the GRIB source name (the first entry) is to invalidate and delete the entire line.

‘Add’ creates a new blank line for user input. ‘Save’ both writes the grib_prep.nl namelist file and updates the pull-down menu seen in the ‘Script’ panel (Figure 16). Note that ‘Save’ is automatically invoked when ‘Script’ tab is pressed. ‘Reload’ reloads the current EXT_DATAROOT/extdata/static/grib_prep.nl

3.3.2 Script Panel

This section of the GUI assists a user to configure their Perl *grib_prep.pl script*, which runs the Fortran executable grib_prep (which in turn reads the file grib_prep.nl).

First, the Script panel assists the user in composing command line arguments for grib_prep.pl. By choosing a Data Source, for example AVN, a command is created in an editable text window shown at the bottom of the panel. A list of command line option switches is shown on the panel resulting from running grib_prep.pl with the help option (grib_prep.pl -h). Second, the user is able to immediately run this command by pressing ‘Run’, or save the script to a file with ‘Save’. (See Figure 16.) If Save is pressed the script will be located in INSTALLROOT/user_scripts available to be run at a later time, or in the case of real-time runs used as input for a cron file.

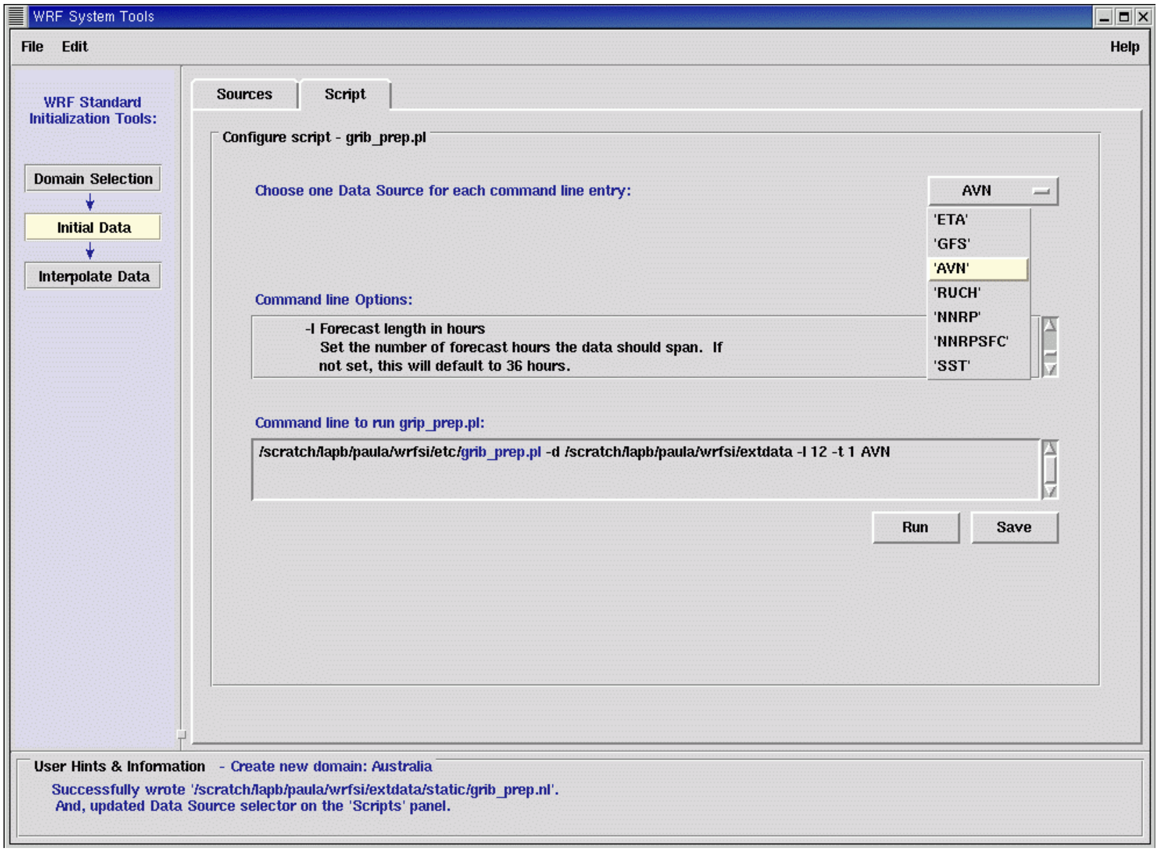

Figure 16: Display of Initial Data Tool - Script panel.

The grib_prep.pl script accepts only one data source at a time. Thus, an additional grib_prep.pl command needs to be created for each GRIB source to process. Please note that the values entered in the Interpolate Data Tool in the 'Controls' panel (Section 3.4.1, Figure 17) need to correspond to choices made here, specifically the values for INIT_ROOT and LBC_ROOT need to be considered. For instance, if INIT_ROOT='AVN' then grib_prep.pl needs to run with the argument AVN. A default of –t 1 is an option that instructs the script to create an hourly data file.

3.4 Interpolate Data Tool

The Interpolate Data Tool (Figure 17) performs the second of two tasks in providing the initial and lateral boundary condition data files. This section of the user interface runs a script (wrfprep.pl), which interpolates background data to a specific domain. In fact, the button to invoke this tool is disabled and appears gray unless a newly created domain is localized or an existing localized domain is loaded via the Domain Selection Tool.

3.4.1 Controls Panel

The Controls panel (Figure 17) contains parameters to interpolate initial and lateral boundary condition files to the domain. The values are written to the namelist and are used by Perl scripts to prepare initial and lateral boundary file discussed in sections 3.3 Initial Data Tool.

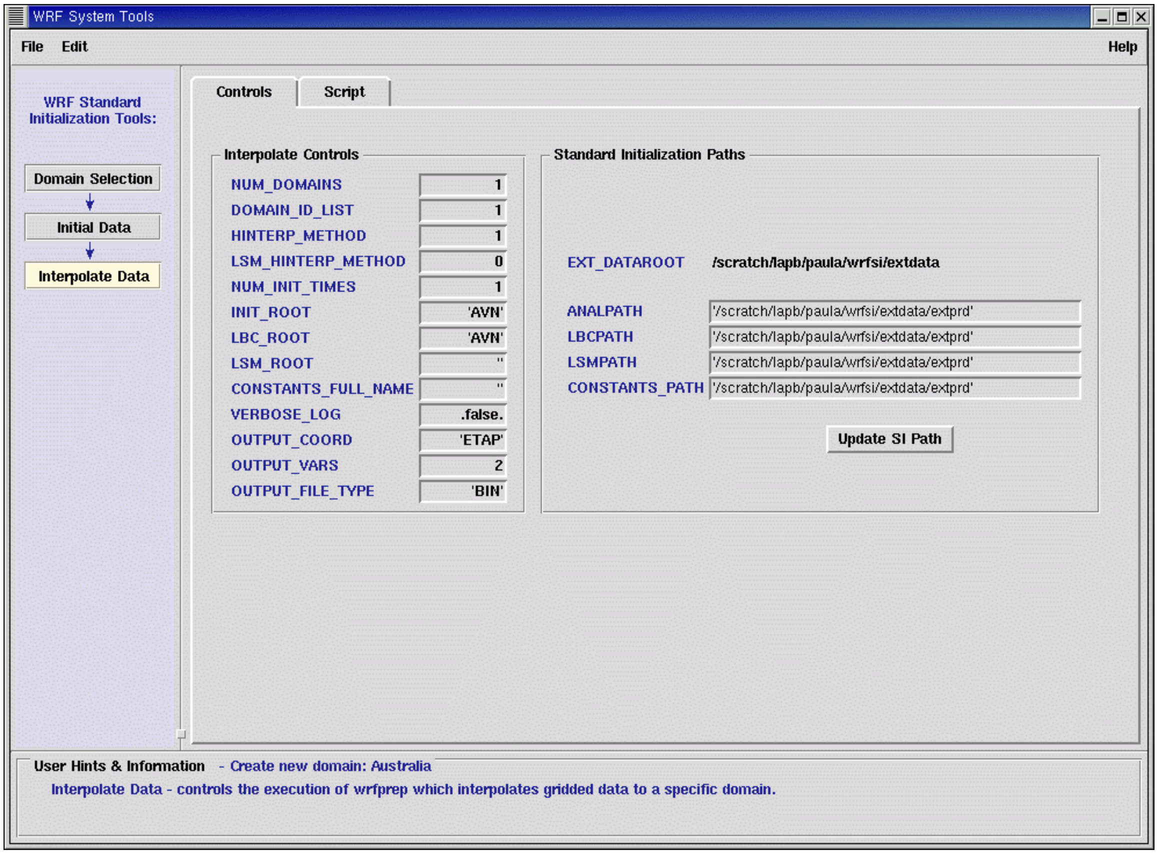

Figure 17: Display of Interpolate Data - Controls panel.

3.4.2 Script Panel

This section of the GUI assists a user to configure their *wrfprep.pl script*.

As can be seen in the user interface (Figure 18) this Script panel assists the user in adding arguments to wrfprep.pl. A list of command line option switches are shown on the panel resulting from running wrfprep.pl the help option (wrfprep.pl -h). The user is able to immediately run this command by pressing ‘Run’, or to save the script to a file with ‘Save’. If Save is pressed the script will be located in INSTALLROOT/user_scripts available to be run at a later time, or in the case of real-time runs used as input for a cron file.

A text window labeled ‘Data files’ lists the data files found in ANALPATH, LBCPATH, LSMPATH and CONSTANTS_PATH created by running grib_prep.pl in the Initial Data Tool (Section 3.3.2, Figure 16).

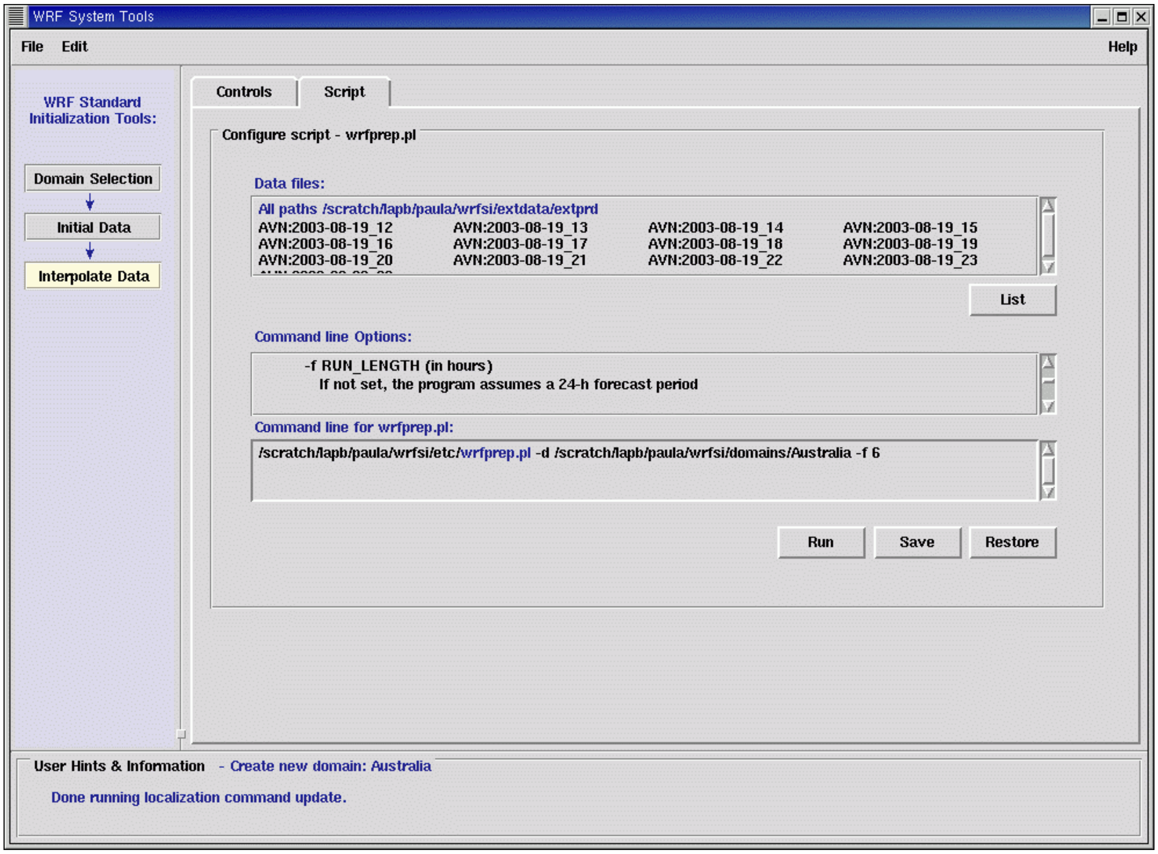

Figure 18: Display of Interpolate Data - Script panel.

The time option (-t) and forecast length (-f) switches used with grib_prep.pl directly relate to the option switches needed for wrfprep.pl to generate interpolated data. To illustrate, if you were to run grib_prep using -t 3 for the ETA files, you would get 12Z, 15Z, etc. If you were to run wrfprep.pl with no time arguments, it would run for the current hour (let's say, 16Z), which requires 16Z, 19Z, etc. for the ETA data. The result would be no interpolated files.

To generate desired file in this illustration, you could do one of many things:

1. Run grib_prep with -t 1, so you get hourly ETA output, thereby satisfying our ability to run the WRF for any hour.

-OR-

2. Use either -o or -s with wrfprep to start the model at a time for which you have some ETA data. Since you, to continue the example, executed the scripts at 16Z, grib_prep determined that the 12Z ETA was the latest run and created output files for 12Z, 15Z, 18Z, etc. You could run wrfprep.pl with -o -4 to say you want the initial hour 4 hours prior to the current hour (16-4 = 12Z). You could also run with -o -1 to use a 15Z initial time.

3.5 Path Preferences

The ability to edit path preferences allows users to enter new paths specifically to external and geographical data from the values that were defined when the install script (install_wrfsi.pl) ran. Changes to the path for external data (EXT_DATAROOT) will require an update to the path choices found in the Standard Initialization Paths in the Interpolate Data in the 'Controls' panel (Section 3.4.1, Figure 17). Changes to the path for geographical data (GEOG_DATAROOT) require an update to the path choices of the Static Geographical Data paths in the Domain Selection Tool in the 'Localization Parms' panel (Section 3.2.4, Figure 9).

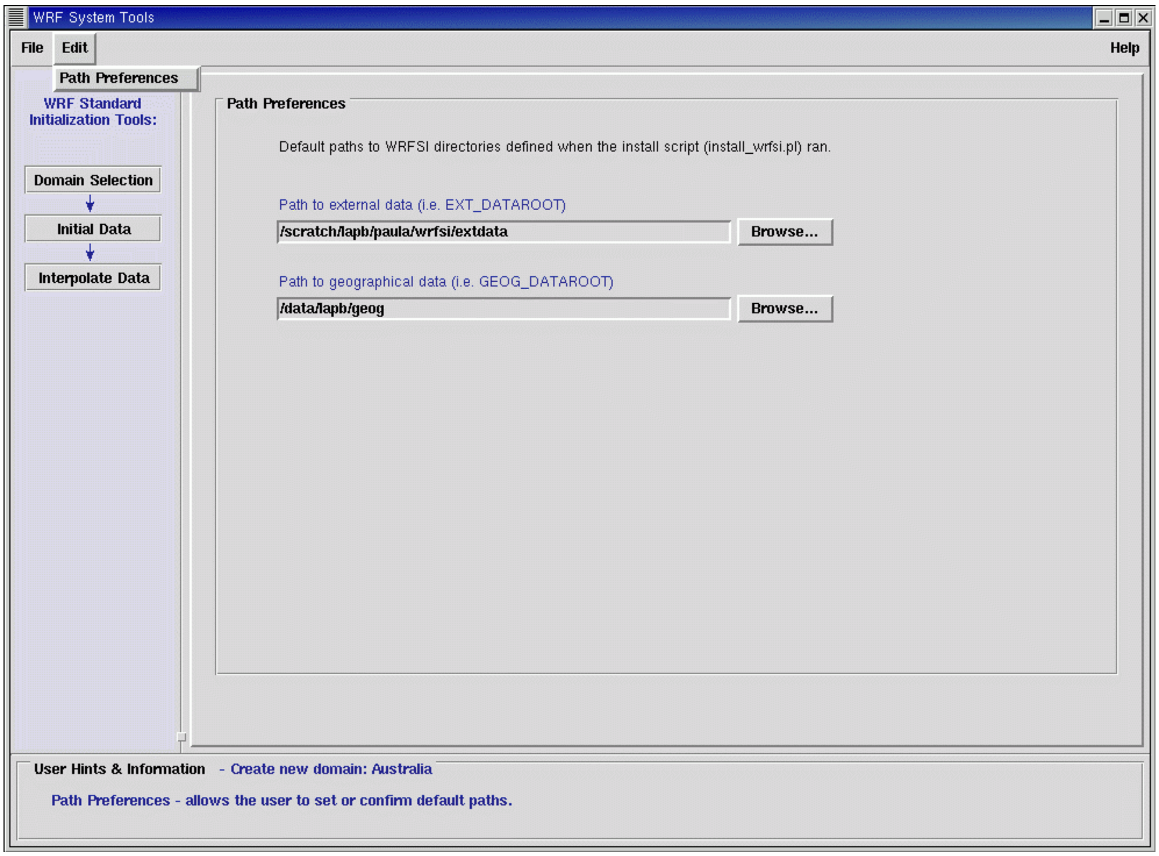

Figure 19: Display of the Path Preferences panel.

The Path Preferences menu button to display the Path Preferences panel is found under the toolbar's Edit pull-down menu.

4.0 Future Work

4.1 Model Setup Tool

The GUI does not yet have a Model Setup tool allowing the user to set up the remaining part of SI but we have plans to incorporate this functionality soon.

4.2 Acquiring Static Geographic Data

Localizing a WRF domain requires geography data files for 7 separate types of static land states information. These are listed in section 3.2.6 as the GUI generates the image displays after domain localization. More information is available in the README on the location, description and ***size*** of these static data sets.

5.0 Summary

The WRF GUI is not a finished product and there are plans for upgrades. On the other hand, the current GUI is capable of doing difficult first steps of numerical weather prediction set up in domain localization and generating interpolated initial and lateral boundary condition files. We envision the GUI to be capable of assisting the user with many other tasks associated with WRF system set up, running and evaluation.

6.0 References (incomplete at the moment)

LIDI02 Lidie, Steve, and Nancy Walsh. Mastering Perl/Tk. Sebastoopol (CA): O'Reilly & Associates, 2002.

JOHN00 Johnson, Jeff. GUI Bloopers: don'ts and do's for software developers and Web designers. San Diego: Academic Press, 2000.